

AP173HO

**APPLICATION
FOR
UNITED STATES LETTERS PATENT**

Be it known that I, Wai Yim, of 2283 Latham Street, Apt. #12, Mountain View, California 94040, a citizen of U.S.A. have invented new and useful improvements in:

CONNECTIONLESS TCP/IP DATA EXCHANGE

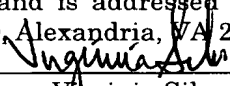
of which the following is the specification

CERTIFICATION UNDER 37 C.F.R. 1.10

"Express Mail" Mailing Label Number: EV311301954US

Date of Deposit: October 8, 2003

I hereby certify that this patent application is being deposited with the United States Postal Service on this date in an envelope as "Express Mail Post Office to Addressee" service under 37 C.F.R. 1.10 on the date indicated above and is addressed to Mail Stop Patent Application, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.



Virginia Silva

CONNECTIONLESS TCP/IP DATA EXCHANGE

by Inventor:

Wai Yim

BACKGROUND OF THE INVENTION

5 **1. Field of the Invention**

[0001] The present invention relates generally to the transmission of information across the Internet, and more specifically to methods, systems, and apparatus for rapid, real-time transmission of information across the Internet and within networks and networked systems.

10 **2. Description of the Related Art**

[0002] Many Internet based applications require rapid transmission and exchange of data for effective implementation. By way of example, H323 Internet video conferencing provides rapid, real time data exchange to present video and audio data for participants in local and remote settings. Typically, to realize the benefits of necessary rapid data exchange, data is transmitted over unreliable User Datagram Protocol/Internet Protocol (UDP/IP). The advantage of using the unreliable UDP over the reliable Transmission Control Protocol (TCP, also TCP/IP) is primarily an advantage of speed. UDP has less overhead since it does not transmit packet acknowledgement, packet verification, requests for packet re-transmission, etc. In real time media transmission and play-back, such transmissions and verification processes impact the overall system performance severely.

[0003] TCP serves as essentially the standard for most Internet data transmission. TCP maintains the highest degree of reliability by ensuring all data is received, received in the correct order, and that the data received is accurate and consistent with the data that was

transmitted. In many applications, such reliability is paramount for effective data transmission. The highest degree of reliability, however, is not necessary for all Internet data transmission. In such applications as H323 Internet video conferencing, for example, speed is paramount. Most applications can easily compensate for occasionally
5 missed audio data, which is generally imperceptible, and similarly, occasionally missed or garbled video data is generally easily tolerated and of little hindrance to video conferencing sessions.

[0004] Figure 1 is a system schematic 10 of an exemplary video conferencing system arrangement illustrating various data exchange paths. Participants 12 exchange audio,
10 video, and other media data, with each other, and often with a video conferencing or other data server 14. In such an arrangement, data exchange can be peer-to-peer 16, client-server 18, and various combinations thereof. In a typical Internet based arrangement, initial set-up and control data such as client set-up, parameter and capability for exchange, and the like, is established using TCP, and then video conferencing media
15 would be exchanged using UDP. For example, during set-up using TCP, a port or range of ports may be designated for UDP data exchange, and then conferencing is conducted using UDP and the designated port or range of ports.

[0005] In the environment of required network and Internet security, many firewalls block or deny all incoming Internet traffic except TCP/IP. Figure 2 shows system
20 schematic 10 illustrated in Figure 1 with the added features of participant firewalls 20 and data server firewall 22. In an environment in which participants 12 are all within discreet networks or locations, data exchange might be across one or more firewalls 20, 22 whether that exchange is peer-to-peer 16 or client-server 18.

[0006] When rapid, real-time transmission is desired, a firewall can and often does limit or prevent desired video conferencing capability. If a particular firewall blocks or denies all incoming Internet traffic except TCP/IP, video conferencing or other data exchange must be conducted using highly reliable, but much slower, TCP/IP, or some work-around must be established to conduct UDP data transmission and exchange. Typical solutions include designating specific ports during set-up for UDP data exchange. By way of example, an H323 session may be established with any port greater than 1028 designated for UDP data exchange. Some firewalls are designed and implemented having certain ports designated for media exchange and allowing UDP data, and some firewalls block all UDP and TCP ports except for TCP port 80, which is the designated HyperText Transfer Protocol (HTTP) port. Problems with such configurations include the lack of standardization across the various kinds and types of firewalls available, and that such work-arounds ultimately defeat the security of the firewall.

[0007] In view of the foregoing, what is needed is a method and system of Internet data exchange with the access of TCP/IP packet transmission, and with the speed and function of UDP data transmission.

SUMMARY OF THE INVENTION

[0008] Broadly speaking, the present invention fills these needs by providing a method and communication protocol for data exchange providing the access to traverse network divisions, firewalls, and the like, of TCP, and the speed of UDP. The present invention
5 can be implemented in numerous ways, including as a process, an apparatus, a system, a device, a method, or a computer readable media. Several embodiments of the present invention are described below.

[0009] In one embodiment, a method of conducting a communication exchange between systems over a communication network is provided. The method includes the formatting
10 of data by a first system into an IP datagram with an IP header and one of a TCP and a UDP header. A connectionless TCP/IP header is constructed to add to the formatted data. The connectionless TCP/IP header includes a pre-defined identifying value in a designated field. The method the provides for transmitting the formatted data having the connectionless TCP/IP header from the first system to a second system. The pre-defined
15 identifying value in the designated field is verified to identify the connectionless TCP/IP header. The method then includes removing the identified connectionless TCP/IP header from the IP datagram, and processing the IP datagram.

[0010] In another embodiment, a method of communication between cooperating systems in a video conferencing system is provided. The method includes constructing a
20 connectionless TCP/IP header. The connectionless TCP/IP header includes a flag in a designated field and a checksum in another designated field identifying the connectionless TCP/IP header. The method next provides for attaching the connectionless TCP/IP header to an IP datagram, and then transmitting the IP datagram

with the connectionless TCP/IP header. The connectionless TCP/IP header is removed by a receiving cooperating system, and the IP datagram is processed without transmitting acknowledgement and without requesting verification.

[0011] In a further embodiment, a communication protocol for establishing and
5 maintaining an exchange between cooperating systems is provided. The communication protocol provides for formatting data to be transmitted into an IP datagram, and for attaching a connectionless TCP/IP header to the IP datagram. The communication protocol further provides for transmitting the IP datagram with the connectionless TCP/IP header as a new IP datagram. The new IP datagram is received and identified as a
10 connectionless TCP/IP header. The communication protocol further provides for removing the connectionless TCP/IP header from the new IP datagram, and then processing the new IP datagram. The new IP datagram is processed without acknowledgement and without transmitting a request for verification.

[0012] In a yet another embodiment, an integrated circuit chip for exchanging
15 communication between cooperating systems is provided. The integrated circuit chip includes logic for constructing a connectionless TCP/IP header. The connectionless TCP/IP header includes a flag in a designated field identifying a communication as a connectionless TCP/IP communication. The integrated circuit chip further includes logic for constructing a checksum to verify the communication is a connectionless TCP/IP
20 communication.

[0013] In another embodiment, a computer readable media having program instructions for exchanging communication between cooperating systems is provided. The computer readable media includes program instructions for constructing a connectionless TCP/IP

header. The connectionless TCP/IP header has a flag in a designated field, and a checksum in another designated field, identifying the connectionless TCP/IP header. The computer readable media includes further program instructions for attaching the connectionless TCP/IP header to an IP datagram, and for transmitting the IP datagram
5 with the connectionless TCP/IP header. The connectionless TCP/IP header is removed by a receiving cooperating system and the IP datagram is processed without transmitting acknowledgement and without requesting verification.

[0014] The advantages of the present invention over the prior art are numerous. One notable benefit and advantage of the invention is that real-time audio and video can be
10 transmitted and exchanged across firewalls. The connectionless TCP/IP header contains valid and credible information in the various fields of the header without consuming optional field space or reserved bits. The high speed data transmission of UDP is therefore achieved, but with the access of TCP/IP. Other advantages of the invention will become apparent from the following detailed description, taken in conjunction with the
15 accompanying drawings, illustrating by way of example the principles of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

[0015] The accompanying drawings, which are incorporated in and constitute part of this specification, illustrate exemplary embodiments of the invention and together with the description serve to explain the principles of the invention.

5 [0016] Figure 1 is a system schematic of an exemplary video conferencing system arrangement illustrating various data exchange paths.

[0017] Figure 2 shows the system schematic illustrated in Figure 1 with the added features of participant firewalls and a data server firewall.

[0018] Figure 3 is an essentially standard IP header.

10 [0019] Figure 4 is an essentially standard TCP header.

[0020] Figure 5 is a connectionless TCP header in accordance with one embodiment of the present invention.

[0021] Figure 6 shows a flow chart diagram of the method operations performed by a sender to construct a connectionless TCP/IP header for data transmission, in accordance
15 with one embodiment of the present invention.

[0022] Figure 7 is a flowchart diagram illustrating the method operations performed by a receiver to determine whether a datagram includes a connectionless TCP/IP header, in accordance with one embodiment of the present invention.

[0023] Figure 8 is a flow chart diagram illustrating the method operations performed to
20 effect data exchange using a connectionless TCP/IP datagram, in accordance with one embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0024] An invention for a communication protocol, as well as for a method and system of data exchange is described. In the following description, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be understood, however, to one skilled in the art, that the present invention may be practiced without some or all of these specific details. In other instances, well known process operations have not been described in detail in order not to unnecessarily obscure the present invention.

[0025] Due to security concerns, the firewall is a part of essentially most systems and networks with Internet access, and is a valuable tool for safeguarding data and maintaining system integrity. Security comes at a price, however, and one price in the area of data exchange over the Internet is speed. While every firewall has its own characteristic methods for establishing and maintaining a desired level of security, it is common for firewalls to deny or block access to all but TCP/IP transmission, or to designate only certain ports or a range of ports for TCP data exchange, for UDP data exchange, or for some firewalls, for identified media exchange. Embodiments of the present invention provide for data transmission in IP datagram packets with a unique media exchange TCP header. The special media exchange TCP header looks very much like a typical TCP header, except that the special media exchange TCP header packages an essentially typical UDP datagram, and therefore the receiver does not send an acknowledgement packet, the sender does not wait for an acknowledgement before sliding the window, and the data exchange is essentially as if using UDP with a TCP header. As used herein, the special media exchange TCP header and associated data

exchange is referred to as a connectionless TCP/IP header, a connectionless TCP/IP datagram, packet, etc.

[0026] Figure 3 is an essentially standard IP header 100. Embodiments of the present invention provide for specific information to be written to particular fields in both the IP header 100, and the TCP header described below, to create a connectionless TCP/IP header. For completeness, each field in the IP header 100, as well as in the TCP header described in Figure 4, is identified and generally described.

[0027] In the IP header 100 shown in Figure 3, Version 102 field is 4 bits in length and indicates the format of the Internet header. Header Length 104 is also 4 bits in length and describes the Internet header length in 32-bit words. The header length 104 in 32-bit words points to the beginning of the data 128. Type of Service 106 is an 8-bit field and indicates the quality of service desired. Some networks offer service precedence and other quality of service options for times of high traffic, service load, etc. Total Length 108 is a 16-bit field that indicates the total length of the datagram in bytes. Total length 108 includes the Internet header or headers and data. Identification 110 field is a 16-bit field containing an identifying value assigned by the sender to aid in assembling the fragments of a datagram. Flags 112 is a 3-bit field containing control flags regarding fragmentation of the datagram. Fragment Offset 114 is a 13-bit field that indicates where the instant fragment may fall in the entire datagram and is usually expressed in bytes. Time to Live (TTL) 116 is an 8-bit field that indicates the maximum number of routers through which the IP datagram can pass. Protocol 118 is an 8-bit field indicating the next level protocol used in the data portion of the Internet datagram. Checksum 120 is a 16-bit field having a checksum on the header only. Source IP address 122 is a 32-bit field

indicating the address of the datagram source, and Destination IP address 124 is a 32-bit field indicating the address of the datagram destination or intended recipient. Options 126 may or may not appear in a datagram, and the field is variable in length. Finally, Data 128 includes the data or a next level protocol header.

5 [0028] Figure 4 is an essentially standard TCP header 130. As described above in reference to the IP header 100 illustrated in Figure 3, embodiments of the present invention provide for specific information to be written to particular fields in both the IP header 100 (see Figure 3), and the TCP header 130, to create a connectionless TCP/IP header. For completeness, each field in the TCP header 130 in Figure 4 is identified and
10 generally described.

[0029] Source port number 132 is a 16-bit field containing the source port number, and Destination port number 134 is a 16-bit field containing the destination port number. Sequence number 136 is a 32-bit field containing the first data octet in this segment, unless SYN is present. If SYN is present, the sequence number 136 is the initial
15 sequence number and the first data octet. Acknowledgement number 138 is a 32-bit field that, if the ACK control bit is set, contains the value of the next sequence number which the sender of the segment is expecting to receive. Header Length 140 is a 4-bit field indicating the number of 32-bit words in the TCP header 130. Header length 140 points to the beginning of data 164. Reserved 142 field is a 6-bit field that must be set to zero.
20 Fields URG 144, ACK 146, PSH 148, RST 150, SYN 152, and FIN 154, together comprise 6 bits and are control bits. Window size 160 is a 16-bit field indicating the number of data octets which the sender of this segment is willing to accept, which defines the window size, framing the data exchanged. Checksum 158 is a 16-bit field containing

a checksum for the TCP header 130. Urgent pointer 160 is a 16-bit field communicating the current value of the urgent pointer as a positive offset from the sequence number in this segment. The field is only applicable to headers in which the URG 144 control bit is set. Options 162 may or may not be present, have a length of a multiple of 8 bits, if present, and are included in the checksum 158. Finally, data 164 is a field of variable
5 length containing the data, or a next level protocol header.

[0030] As is well known, the IP is defined by standard, and the IP header 100 (see Figure 3) is the routing identifier and instructions used to route datagrams from host to host. TCP, also defined by standard, essentially provides the reliability to the datagram
10 transmission through acknowledgement, verification, and re-transmission where appropriate. It should therefore be understood that, in order to create and transmit with a connectionless TCP/IP header, the data fields of the header need to be in accordance with the corresponding TCP/IP standards, and that credible, valid, information must be contained in the appropriate data fields of the header.

15 [0031] In one embodiment of the present invention, a connectionless TCP/IP header is created by defining data values to be transmitted in the fields corresponding to the Window size 156 shown in Figure 4. No additional or extra bits are introduced, and none of the reserved bits are used. In one embodiment, the window size 156 (see Figure 4) field is used to define a connectionless TCP/IP header. Since no acknowledgement is
20 transmitted in data exchange using a connectionless TCP/IP header, the window size 156 field is of little relevance to the data exchange in embodiments of the present invention. The 16-bit window size 156 field is sub-divided into an upper byte and a lower byte. In one embodiment of the invention, a pre-defined value is written to the lower byte of the

16-bit window size 156 field to identify the connectionless TCP/IP header. In other words, the pre-defined value written to the lower byte of the window size 156 field identifies the datagram as a connectionless TCP/IP transmission, differentiating the datagram from standard TCP/IP transmissions. In other embodiments, the pre-defined
5 value can be written to the upper byte of the window size 156 field.

[0032] In addition to the pre-defined value in the lower byte of the window size 156 field, in one embodiment, the upper byte of the window size 156 field shall carry a special checksum, described in greater detail below, to verify and confirm that the datagram is a connectionless TCP/IP datagram. In an embodiment in which the pre-
10 defined value is written to the upper byte of the window size 156 field, the special checksum is written to the lower byte of the window size 156 field. In one embodiment, if the pre-defined value in the lower byte of the window size 156 field identifies the received packet as a connectionless TCP/IP datagram, and the checksum in the upper byte of the window size 156 field of a connectionless TCP/IP header validates the
15 identification, the datagram will be treated and processed as a connectionless TCP/IP transmission.

[0033] Figure 5 is a connectionless TCP header 170 in accordance with one embodiment of the present invention. As described above, the connectionless TCP header 170 is essentially the same as a regular TCP header 130 (see Figure 4), with the above described
20 modification to the window size field. As shown in Figure 5, the window size field has been subdivided into an upper byte 172 and a lower byte 174. All remaining fields are essentially identical to the regular TCP header 130 as illustrated in Figure 4. In one embodiment of the invention, a checksum is written to the upper byte 172 of the window

size field, and a pre-defined value is written to the lower byte 174 of the window size field. The pre-defined value and the checksum identify the datagram as a connectionless TCP/IP datagram. In other embodiments, the pre-defined value and the checksum are written to the upper and lower bytes of the window size 156 field, respectively.

- 5 [0034] Actions of a sender and a receiver of a connectionless TCP/IP transmission are used to illustrate data exchange using a connectionless TCP/IP header. Figure 6 shows a flow chart diagram 200 of the method operations performed by a sender to construct a connectionless TCP/IP header for data transmission, in accordance with one embodiment of the present invention. The method begins with operation 202 in which a pre-defined
- 10 value is written to the lower byte of the window size field of an otherwise typical TCP standard header block to identify a connectionless TCP/IP datagram. A typical TCP standard header block is illustrated in Figure 4, and a connectionless TCP header is shown in Figure 5. In one embodiment, the pre-defined value can be any number between 0 and 225, and in one embodiment the pre-defined number is an odd number.
- 15 [0035] The method continues with operation 204 in which the two bytes of the Identification value in the IP header are summed. A typical IP standard header is illustrated in Figure 3. In an embodiment of the present invention, the identification field must have valid data to be recognized as a proper IP header. In operation 204, the existing value is summed, and the result will be used to further verify that the identified
- 20 connectionless TCP/IP header is in fact a connectionless TCP/IP header. In other embodiments of the invention, any desired valid and reliable field or combination of fields can be used to calculate the checksum.

[0036] Next, in operation 206, a one's complement calculation is performed on the result of operation 204. A one's complement calculation is a common checksum calculation used in various operations and fields in datagrams. In an embodiment of the present invention, a one's complement calculation utilizes an existing value in a header field, and
5 thereby defines a field to verify the connectionless TCP/IP header.

[0037] The method concludes with operation 208 in which the one's complement result from operation 206 is written to the upper byte of the window size field of the TCP header. The upper byte of the window size field then contains the calculated one's complement checksum, and the lower byte contains the pre-defined value written in
10 operation 202. In other embodiments, the pre-defined value may be written to the upper byte of the window size field, and the checksum may be written to the lower byte of the window size field. As will be described below in reference to Figure 7, the connectionless TCP/IP header is constructed to present as a valid TCP/IP header, but before acknowledgement or other transmission is conducted, the header is examined to
15 determine if the data is formatted in a connectionless TCP/IP header. With the writing of the one's complement result in operation 208, the method is done.

[0038] Recognition of a connectionless TCP/IP header by a receiver or receiving application is necessary to realize the desired UDP-type processing of the datagram. Figure 7 is a flowchart diagram 220 illustrating the method operations performed by a
20 receiver or receiving application to determine whether a datagram includes a connectionless TCP/IP header, in accordance with one embodiment of the present invention.

[0039] The method begins with operation 222 in which the lower byte of the window size field of the TCP portion of the connectionless TCP/IP header is read. As described above in reference to flowchart diagram 200, a first identifier of a connectionless TCP/IP header is a pre-defined value written to the lower byte of the window size field. In operation
5 222, a first step by the receiver is to read the lower byte of the window size field.

[0040] The method continues with decision block 224 in which it is determined whether the value read from the lower byte of the window size field in operation 222 is a pre-defined value identifying the header as a connectionless TCP/IP header. As will be described in greater detail below, the value can be any number between 0 and 225, and in
10 one embodiment the value is an odd number.

[0041] The method continues with decision block 224 in which it is determined if the value in the lower byte of the window size field is the pre-determined value. If the value is not the pre-determined value, a "no" to decision block 224, the method proceeds to operation 226 in which the datagram is assumed to be a regular TCP/IP datagram and is
15 processed as a TCP/IP datagram, and the method is done. If the value in the lower byte of the window size field is the predetermined value, a "yes" to decision block 224, the method continues with operation 228.

[0042] In operation 228, the upper byte of the window size field is summed with the IP Header identification field, and the carry-over bit is dropped. In other words, the result of
20 the sum of the upper byte of the window size field and the IP Header identification field is truncated to a byte. In decision block 230, it is determined whether the result is "FF." If the result is "FF," a "yes" to decision block 230, the method proceeds to operation 232 in which the datagram is processed as a connectionless TCP/IP datagram, and the method

is done. If, in decision block 230, it is determined the result is not "FF," a "no" to decision block 230, the method loops back to operation 226 in which the datagram is processed as a regular TCP/IP datagram, and the method is done.

[0043] In one embodiment of the invention, a connectionless TCP/IP header is constructed and transmitted to effect essentially UDP data exchange with the access of TCP/IP. As described above in reference to Figures 6 and 7, a connectionless TCP/IP header is constructed for data transmission. The data to which the connectionless TCP/IP header is attached is a UDP datagram. In one embodiment of the invention, the processing of a connectionless TCP/IP datagram includes stripping the connectionless TCP/IP header from the datagram to which it is attached, and then processing the underlying datagram in accordance with whatever the protocol for the underlying datagram may be. For audio and video exchange in a video conference, for example, the audio and video data might be formatted as a UDP datagram. A connectionless TCP/IP header is constructed, and attached to the audio and video UDP datagram. A sender transmits the connectionless TCP/IP datagram, and the receiver receives the connectionless TCP/IP datagram. As the receiver processes the connectionless TCP/IP datagram, it is identified as a connectionless TCP/IP datagram. The connectionless TCP/IP header is stripped from the underlying datagram which is processed according to its own protocol, which in this example is the UDP audio and video data. The data is processed according to UDP protocol, and so no acknowledgement, request for retransmission, and so forth is transmitted. The reliability of TCP/IP is discarded, but the access of cross-firewall transmission is retained, and the speed of UDP is realized.

[0044] In one embodiment of the invention, the lower byte of the TCP window size field (174 in Figure 5) contains a pre-defined value to indicate the header is a connectionless TCP/IP header. The assigning of a specific, pre-defined value to the lower byte 174 of the window size field essentially reduces the probability of mis-identification to approximately 1/125.

[0045] The pre-defined value written to the lower byte 174 of the window size field can be any number between 0 and 225, and in one embodiment, the pre-defined value is an odd number. It should be appreciated that most window size values in standard TCP/IP headers are multiples of 2^X , which is an even number. By choosing an odd number, the chance of mis-identifying a connectionless TCP/IP header is drastically reduced.

[0046] Further, having a checksum of the 2 bytes of the IP header field (110 of Figure 3), and having the checksum define the value of the upper byte of the window size field (172 in Figure 5), results in a probability of mis-identification of a connectionless TCP/IP to approximately $1/256 * 1/256$. The overall probability of misidentifying a connectionless TCP/IP header therefore is approximately $1/256 * 1/256 * 1/256$, or 0.0000059604644775390625%. It should be appreciated that, even if a regular TCP/IP datagram were mis-identified as a connectionless TCP/IP datagram, the result would be a request for re-transmission of the packet by the original TCP/IP application for which the packet was intended. The receiver, or the receiving application, that received the mis-identified packet typically discards the packet as an erroneous packet. In an H323 video conferencing session, for example, the mis-identified packet would simply be processed as a corrupted packet and would be discarded by the TCP/IP stack or the application itself.

[0047] Figure 8 is a flow chart diagram 250 illustrating the method operations performed to effect data exchange using a connectionless TCP/IP datagram, in accordance with one embodiment of the present invention. The method begins with operation 252 in which a sender formats data into datagrams for transmission. In one embodiment, the data is audio and video data in an H323 video conferencing session. As described above, a video conferencing session is exemplary of Internet data exchange in which speed is paramount, and reliability is less important. UDP/IP data exchange might be an ideal transmission protocol to achieve high speed, with less reliability, but in many corporate Internet environments, for example, UDP protocol is severely restricted or blocked. In embodiments of the present invention, real time media data such as audio and video data is formatted into UDP datagrams in operation 252.

[0048] The method continues with operation 254 in which the sender constructs a connectionless TCP/IP header, which will be attached to the data formatted in operation 254. The construction of the connectionless TCP/IP header includes writing a pre-defined value to a lower byte of the window size field of a TCP header. The pre-defined value can be any number between 0 and 225, and in one embodiment the pre-defined value is an odd number. Next, the two bytes of the identification value in the IP header (see Figure 3) are summed, and a one's complement is calculated from the result. The value of the one's complement is written as a checksum to the upper byte of the window size field in the TCP header, creating a connectionless TCP/IP header. As described above, in other embodiments of the present invention, the checksum may be written to the lower byte of the window size field and the pre-defined value may be written to the

upper byte of the window size field. Additionally, any field or combination of fields having valid and credible data can be used to calculate the checksum.

[0049] Next, in operation 256, the sender attaches the connectionless TCP/IP header to the datagram, and in operation 258, the sender transmits the datagram having the connectionless TCP/IP header to a receiver. When transmitted over the Internet, the connectionless TCP/IP datagram appears as a regular TCP/IP datagram with valid and credible values in all of the fields of the header. The data transmission of the connectionless TCP/IP datagram is therefore at the same precedence and using essentially the same protocol as a regular TCP/IP datagram.

[0050] In operation 260, the connectionless TCP/IP datagram is received by a receiver or receiving application. As is known, in data exchange the roles of sender and receiver alternate and shift among participants, data servers, and the like. In the present example, a sender and a receiver are described and can include any participant in a data exchange functionally serving as a sender or a receiver, including participants, data servers, media servers, etc.

[0051] The method continues with operation 262 in which the receiver identifies the datagram as a connectionless TCP/IP datagram. In one embodiment, the receiver checks the lower byte of the window size field of the connectionless TCP header. If a pre-determined value is identified, the receiver then sums the upper byte of the window size field of the connectionless TCP header with the identification field of the IP header. If, after dropping the carry-over bit from the result, the resulting value is "FF", the receiver has identified the header as a connectionless TCP/IP header to be processed accordingly.

[0052] The method continues with operation 264 in which the processing of the connectionless TCP/IP datagram includes stripping the connectionless TCP/IP header from the datagram. In one embodiment, the stripping of the connectionless TCP/IP header leaves the datagram that was formatted in operation 252. In an embodiment in
5 which the datagram is real time audio and video data from an H323 video conferencing session, the underlying datagram might be formatted as a UDP datagram. An H323 video conferencing session may include other media data, and other data, in which reliability is paramount. In such an example, the underlying datagram might be formatted to utilize highly reliable regular TCP/IP transmission protocol.

10 [0053] In operation 266, the receiver identifies the protocol for the underlying datagram, and in operation 268, the method concludes with the receiver processing the underlying datagram in accordance with the identified protocol. In the example of the connectionless TCP/IP protocol used to transmit UDP real time audio and video data, there is no acknowledgement, verification, or request for retransmission. A receiver, in
15 one embodiment, conducts error correction on the received data, but then any missed, garbled, or erroneous received data is compensated for, and rapid processing of the data is achieved and maintained.

[0054] In summary, embodiments of the present invention enable data exchange of less reliable, but fast, UDP data in a network and Internet security environment in which
20 TCP/IP protocol is increasingly required. A UDP, TCP, ICMP, or any IP encapsulated protocol, datagram is packaged in a connectionless TCP/IP header which appears essentially as a regular TCP/IP header and datagram. Upon receipt, the datagram is identified as a connectionless TCP/IP datagram, the connectionless TCP/IP header is

stripped from the datagram, and the data is processed in accordance with the original or underlying protocol that remains after the connectionless TCP/IP header is removed. The connectionless TCP/IP header does not automatically initiate a response such as an acknowledgement, a verification, a request for re-transmission, and so forth. In one embodiment, the connectionless TCP/IP header enables transmission over the Internet and other networks that may prioritize or require the TCP/IP protocol, but the identification of datagram as a connectionless TCP/IP datagram results in the header being stripped from the underlying datagram and precludes traditional acknowledgement and other reliability data exchange.

- 10 [0055] With the above embodiments in mind, it should be understood that the invention may employ various computer-implemented operations involving data stored in computer systems. These operations are those requiring physical manipulation of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated. Further, the manipulations performed are often referred to in terms, such as producing, identifying, determining, or comparing.

- [0056] The invention can also be embodied as computer readable code on a computer readable medium. The computer readable medium is any data storage device that can store data which can be thereafter read by a computer system. The computer readable medium also includes an electromagnetic carrier wave in which the computer code is embodied. Examples of the computer readable medium include hard drives, network attached storage (NAS), read-only memory, random-access memory, CD-ROMs, CD-Rs, CD-RWs, magnetic tapes, and other optical and non-optical data storage devices. The

AP173HO

computer readable medium can also be distributed over a network coupled computer system so that the computer readable code is stored and executed in a distributed fashion.

[0057] Although the foregoing invention has been described in some detail for purposes of clarity of understanding, it will be apparent that certain changes and modifications may
5 be practiced within the scope of the appended claims. Accordingly, the present embodiments are to be considered as illustrative and not restrictive, and the invention is not to be limited to the details given herein, but may be modified within the scope and equivalents of the appended claims.

What is claimed is: